EE334 **final Exam.** **Time: 180**

==

Q-1……………………………………………………………………………………….[10]

Consider CS = 2000h ; DS = 1500h ; DI = 0100h ; BX = 0130h ; SS = 5000h

SP = 0250h ; BP = 1400h ; AX = 4C69h ; DX = 8855h ; CX= 6632h

(i) Find a physical address of the lower and upper range of CS, SS and DS memory .

Cs : lower address : 20000h upper address : 2FFFFh …………..... [1.00].
SS : lower address : 50000h upper address : 5FFFFh …..…...... [1.00].
DS : lower address : 15000h upper address : 24FFFh ……......... [1.00].

(ii) Find the logical and physical address of a data stored at offset address BX .

Logical address : 1500:0130 physical address : 15130h …………..… [0.5]

(iii) Find the logical and physical address of a data stored at offset address BP.

Logical address : 5000:1400 physical address : 51400h ………… [0.5]

(iv) Find the contents of destination and the physical address accessed by each of the following instructions:

- MOV [Bp+100], AX
  Physical address : 51500h , contain 4C69h …………………..… [1.00].
- MOV SS: 5[BX][DI], DX
  physical address : 50235h , contain 8855h ………..…….…..... [1.00].
- MOV [0200h], CL
  Physical address : 15200h , contain 32 ……………………....... [1.00].

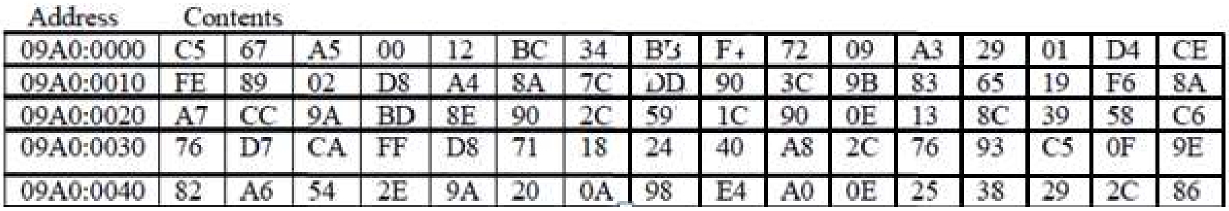
(v) What are The addressing mode have used in each of the above instruction.

based relative …………………………………..……………....... [1.00].
based indexed relative ………………..……..……………………......... [1.00].
register direct ………………………………………………………...... [1.00].

==

Q-2 ……………………………………………………………………………………[10]

Assume the following memory contents at the start of each of the following instructions



Address	Contents															
09A0:0000	C5	67	A5	00	12	BC	34	B3	F4	72	09	A3	29	01	D4	CE
09A0:0010	FE	89	02	D8	A4	8A	7C	DD	90	3C	9B	83	65	19	F6	8A
09A0:0020	A7	CC	9A	BD	8E	90	2C	59	1C	90	0E	13	8C	39	58	C6
09A0:0030	76	D7	CA	FF	D8	71	18	24	40	A8	2C	76	93	C5	0F	9E
09A0:0040	82	A6	54	2E	9A	20	0A	98	E4	A0	0E	25	38	29	2C	86

1- Assume the following register contents at the START of each of the following instructions. ES: 09A0, DS: 09A0, SS: 09A1 , AX = E265, DX = 73A2, CX = 0000, SP = 0018, BP=002E what is the effectiveness of the following instruction on the register and memory content above?

a. Push ax *sp (new) = 0016, from memory map above (SS:SP = 09A1:0018 = 09A0: 0028) so the location 09A0:0027 contain E2 (AH) , and 09A0:0026 contain 65 (AL) ……………………………………………………………………….......................... [1.00]*

b. sar dl , 2 *dl=E8h …………………………………………………….................... [0.33]*

c. shr dl,2 *dl= 28h ………………………………………………….................... [0.33]*

d. shl al,1 *al = CAh ……………………………………………………………..... [0.33]*

e. xor dl, al *dl = C7h ……………………………………………………………. [0.33]*

f. or dl, F0h *dl = F2h ……………………………………………………….….. [0.33]*

g. not dl *dl = 5Dh ……………………………………………………..….….. [0.33]*

h. pop dx *SP= 18 , after POP execution dx = 901C h , SP (new)=001A . [1.00]*

i. mul dl *ax = al * dl = 101 * 162 = 16362 = 3FEA h .……………………..……... [0.33]*

j. imul dl *ax = ax * dl = 101 * -94 = -9494 = DAEA h …..…………………. [0.33]*

k. and dl, 22 *dl = 22 h ………………………………………………...……. [0.33]*

2- Assume the same register contents/memory contents as above. For EACH of the following two instruction sequences, tell if the jump is TAKEN or NOT TAKEN.

a. cmp al,dl
jne there *TAKEN (al not equal to dl) ………….……………………………..... [0.33]*

b. cmp al,dl
jl there *NOT_TAKEN (al, a postive number, is not less than dl, a negative number) ……………………………………………………………………..... [0.33]*

c. cmp al,dl
ja there *NOT_TAKEN (al is not higher than dl) ……………………………..... [0.33]*

d. cmp ax,dx
jg there *NOT_TAKEN (ax, a negative number is not greater than dx, a postive number)………………………………………………………………. [0.33]*

e. cmp ax,dx
jb there *NOT_TAKEN (ax is not below dx) ……….……………………… [0.33]*

f. test al,1
jnz there *TAKEN (result of al AND 1 is non-zero, so branch taken)…………. [0.33]*

g. add al,dl
jnc there *NOT_TAKEN (al+dl produces a carry, a branch not taken) …….. [0.33]*

h. add al,dl
js there *NOT_TAKEN (MSB of al+dl is '0', so branch not taken) …..…… [0.33]*

i. add al, 40h
jno there *NOT_TAKEN (al + 40h produces an overflow, so branch not taken)[0.33]*

3. Register AL has an 8 bit value (b7b6b5b4b3b2b1b0). Use a single logical instruction to change the contents of AL to (b700000b1b0). Bits B7, B1, B0 unchanged, other bits set to zero.

AND al, 83h ………………………………………………….. [1.00]

4. Register AL has an 8 bit value (b7b6b5b4b3b2b1b0). Use a single logical instruction to change the contents of AL to (1b6b5b4b311b0). Bits B7, B2,B1, set to '1', other bits unchanged.

OR al, 86h …………………………………………………..... [1.00]

==

Q-3 answer any two of three equations below . ……………………………………….[10]

a. Write a subroutine that will return the number of bytes in a string. The string is terminated by a '0'byte (count DOES NOT INCLUDE the '0'byte), and the starting address of the string is passed to the subroutine via the DS:SI register. The count should be returned as zero if the string is 'empty' (first byte is zero). The count should be returned in the AL register (maximum number of characters will be 255).

```
Strlen      PROC
            MOV AX , 00H
    LP1:    MOV BL , [SI]
            CMP   BL , 00H
            JZ    EXIT
```

```
                INC  AL
                INC SI
               JMP LP1
      EXIT:    ret
Strlen         endp
```

Q-3 b . Write a subroutine that will return the maximum 16-bit SIGNED integer from an array of integers. On subroutine entry, register SI will point to the start of the array (each element is 16 bits), and register CX will have the number of integers in the array. The maximum value should be returned in the AX register?

```
           Findmax       proc
                                             ;;; will assume that array always has at least 1 element
                 MOV AX, 8000h                  ;; get most negative 16-bit value into ax
       lp1:    CMP AX , [SI]
               JGE   skip
               MOV AX, [SI]                     ;; ax is less than [SI], get memory value
     skip:    ADD SI , 2                        ;; increment pointer by 2 bytes
               LOOP LP1                         ;; cx has count
              RET
              Findmax          endp
```

Q-3 c. Write a subroutine that will take the lower 4-bits in AL (value is between 0 and 15) and convert it to its ASCII equivalent character representation (value 0 = '0' (30h), value 1 = '1' (31h), etc, value 15 = 'F' (46h). Use capital letters 'A' thru 'F' for values 10 thru 15. The ASCII character value should return in AL. You DO NOT KNOW what the upper 4-bits of register AL contains upon entry to the subroutine?

```
          convert       PROC
                          AND  AL  ,0Fh
                         CMP AL , 09H
                         JG  chr
                         ADD  al  , 30h
                        JMP  exit
     chr :         ADD  AL , 37H
                        exit :
                       RET
   convert      ENDP
```

===

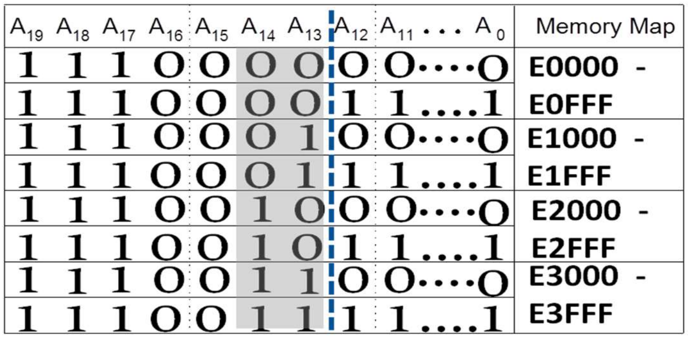
Q-4 ……………………………………………………………………………………………[12]

a .Show how a 32Kbyte ROM module can be connected on an 8088 system using 2764 EPROM chips (8Kbyte), occupying the address range starting from the address E0000H. Use the following address decoding circuits:

1- Nand decoding circuits   2- A line decoder and a Nand gate .

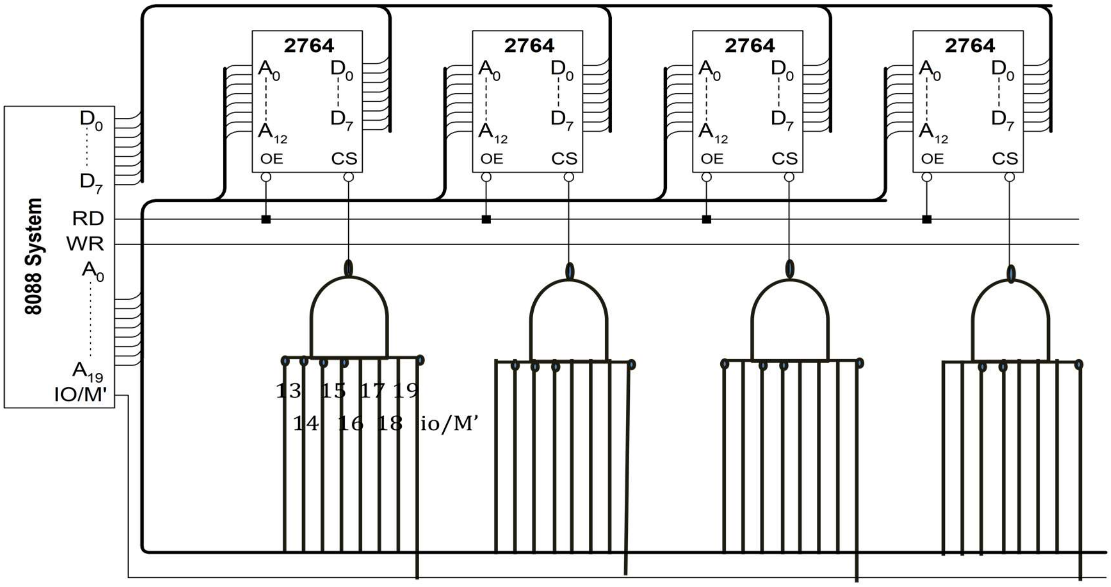
Size of 2764 EPROM chips: 8Kbyte   , Number of chips needed: 4 chip , Number of address lines: 13 address line (A0 –A12)

…………………………………………………………………………………………… [1]

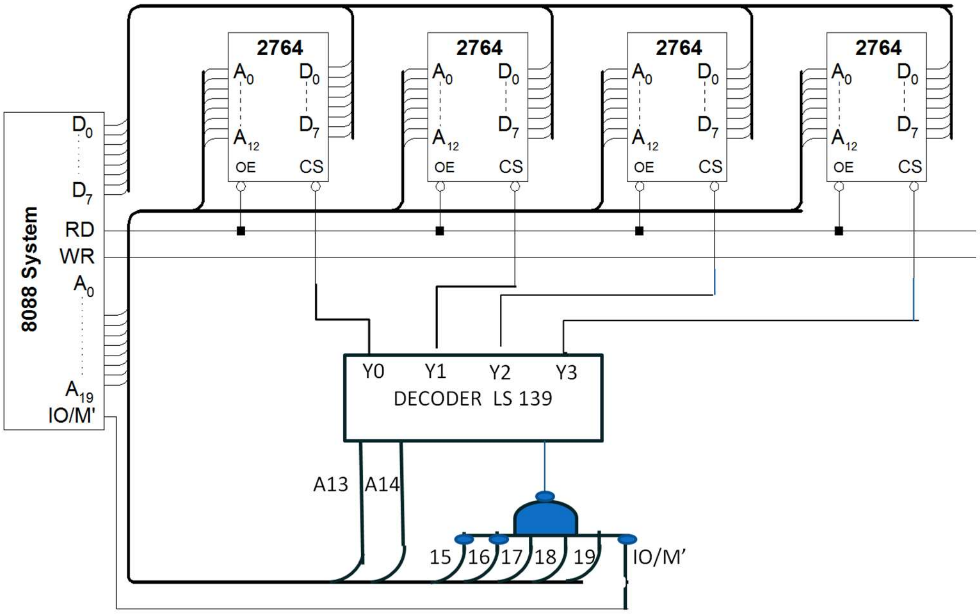


A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11} . . . A_0	Memory Map
1	1	1	0	0	0	0	0	0····0	E0000 -
1	1	1	0	0	0	0	1	1....1	E0FFF
1	1	1	0	0	0	1	0	0····0	E1000 -
1	1	1	0	0	0	1	1	1....1	E1FFF
1	1	1	0	0	1	0	0	0····0	E2000 -
1	1	1	0	0	1	0	1	1....1	E2FFF
1	1	1	0	0	1	1	0	0····0	E3000 -
1	1	1	0	0	1	1	1	1....1	E3FFF

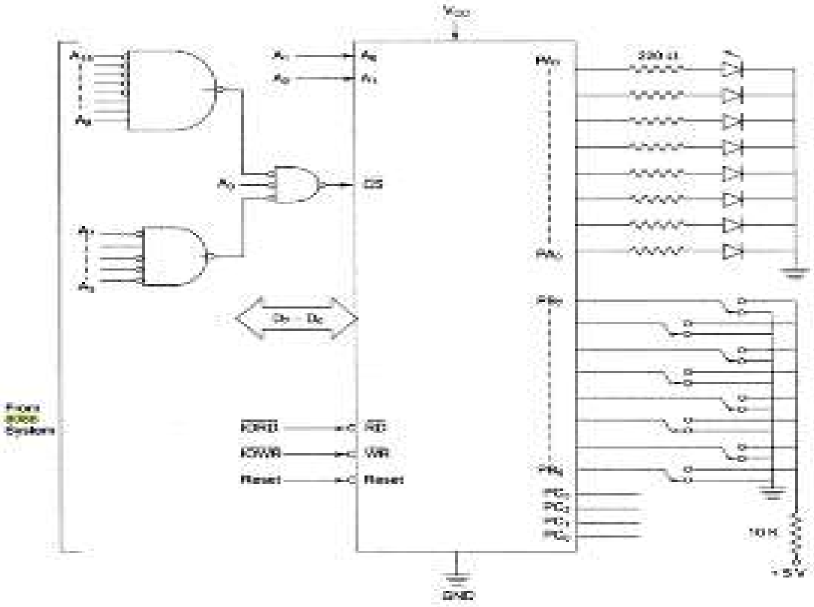
Nand decoding circuits ……………………………………………………………………. [1]



A line decoder and a Nand gate ……………………………………………………………….[1]



Q-4 b. : interface an 8255 with 8086 ? Initialize port A as output port Port B as input and port C lower as output . Port A address should be 0740H. Write program to sense switch positions SW0-SW7 connected to port B . The sensed pattern is to be displayed on port A , to which 8 LEDs are connected?



…………… …….. [2.5]

.data ; ……………………………………………………………………….*[2]*

```
.data
.code
.startup
 mov al , 10000010b
mov dx, 0746h
out dx ,al
mov dx , 0742h
in al ,dx
mov dx , 0740h
out dx , al
ret
end
```

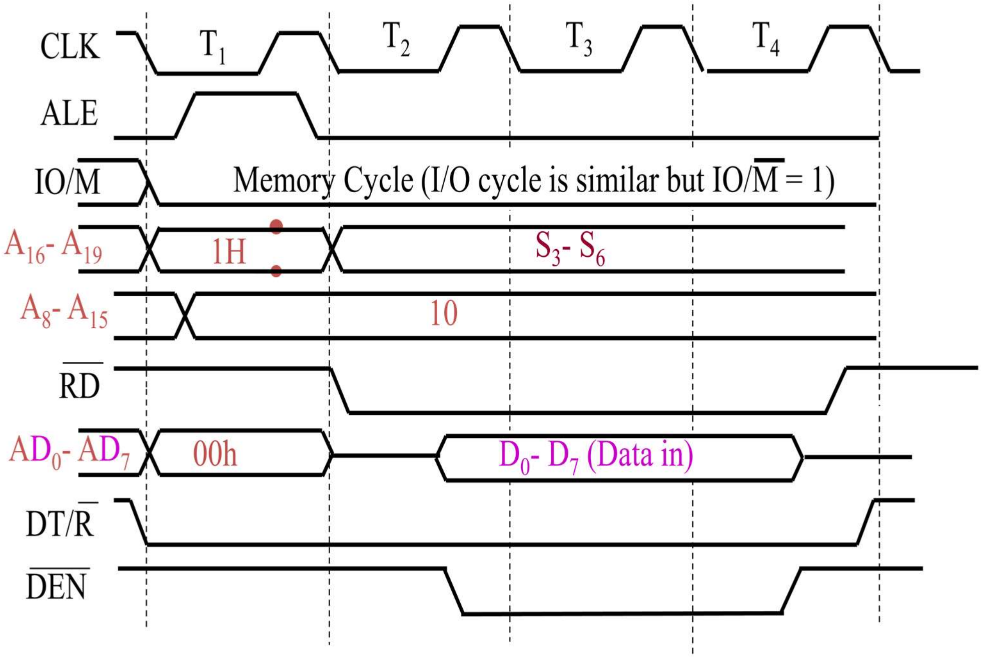
Q-4 c. what is the 8255 A control word register value if : PORT A is a mode 1 input strobe , PORT B is a mode 1 output strobe ,also what is the CWR to enable INTEA and INTEB?

* Note the two unused bits of PORT C output bits

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
	1	0	1	1	0	1	0	0	=	0B4
	Mode Definition	Mode 1		A is input	C is output	mode 1	B is output	C is output		
INTEA	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
	0	0	0	0	1	0	0	1	=	09
	Bit set/reset					Bit 4		set		
INTEB	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
	0	0	0	0	0	1	0	1	=	05
	Bit set/reset					Bit 2		set		

……..*[2.5]*

Q4-d . For 8088 system , draw the timing diagram for the following instruction : Mov AL, [1000] ? ASSUME DS=1000H,



…………[2]

==

Q-5 : ……………………………………………………………………………………..[8]

1- For the 8255A, assume the following port locations: Port A: 1340h, Port B: 1344h, Port C: 1348h, Control: 134Ch. Write an instruction sequence that will configure Port A as an INPUT, Port B as an OUTPUT, Lower half of Port C as an INPUT, and Upper Half of port C as an OUTPUT. Use MODE 0 for all ports.?

P A as an input , P B as an output , P CL as an input , P CU as an output , mode 0

so CWR == 10010001 = 91H

```
MOV  AL ,91H
MOV  DX , 134Ch
OUT  DX  , AL     ……………………[1]
```

a. For 8255A, and assuming the port definitions above, write an instruction sequence that will read a byte from Port A and then write that byte to Port B? …………………………..[2]

read byte from PA == PA is an input

write byte to PB == PB is an output

so CWR == 1001X00X = 90H (assuming X = 0

```
 MOV  AL ,91H
 MOV  DX , 134Ch
 OUT  DX  , AL
MOV DX , 1340h
IN  AL  , DX             ; read byte from PA
```

```
                                        MOV DX , 1344h
                                        OUT DX , AL          ;   write byte to PB
```

b. For Keyboard device was interfaced to the 8086 microprocessor via PPI8255 chip Write a subroutine that will check if any key pressed, and return the character in the AL register along with setting the carry flag = 1. If no data is available, the subroutine should return immediately with the carry flag set = 0 (you cannot assume the value of the carry flag when the subroutine is called). Your subroutine should NOT wait for data to become available.(port B # = 0713h)? ……………………………………………………………[5]

assume the keyboard connected to PB
8255 must be configured in mode 1 and PB must be input strobe
so CWR == 1XXXX11X = 86h

```
                        …………………………...
                         MOV AL , 86h
                         MOV DX , 0717h
                         OUT  DX , AL
                        MOV DX , 0715h
                        IN AL  , DX
                        TEST  AL , 02h
                        JZ  clear-carry
                               MOV DX, 0713h
                               IN  AL , DX
                               STC
                               JMP  exit
                        clear-carry :    CLC
                            exit      :
                        ……………………………..
```

=====================GOOD LUCK ===============================

Q-5 b) solution with an explanation

assume the keyboard connected to PA
8255 must be configured in mode 1 and PA must be input strobe
so CWR == 1011XXXX = B0h
to check if any key is pressed or not we have to check if the IBF (for port A) is active or not (IBFa is active high) , also IBF is a PC5

```
 …………………..
 ; initialize PPI8255
  MOV AL , B0h
  MOV DX , 0717h     ; CWR
  OUT  DX , AL
 ;………………… check IBF active or not
 MOV DX , 0715h   ; Port C
 IN AL  , DX           ;  read Port C
 TEST  AL , 40h       ; check PC5 (IBFa) = = 1 or not
 JZ  clear-carry          ;  if IBFa (PC5) = 0 (de-active) will jump to clear carry
    MOV DX, 0711h
    IN  AL , DX        ; read the pressed key from PA
    STC                ;  set carry flag
   JMP  exit
clear-carry :    CLC  ;   clear carry flag
      exit      :
          ………………
```

another solution ;

assume the keyboard connected to PB

8255 must be configured in mode 1 and PB must be input strobe

so CWR == 1XXXX11X = 86h

to check if any key is pressed or not we have to check if the IBF (for port B) is active or not (IBFb is active high) , also IBF is a PC1

```
 …………………..
 ; initialize PPI8255
  MOV AL , 86h
  MOV DX , 0717h     ; CWR
  OUT  DX , AL
 ;………………… check IBF active or not
 MOV DX , 0715h   ; Port C
 IN AL  , DX           ;  read Port C
 TEST  AL , 02h      ; check PC1 (IBFb) = = 1 or not
 JZ  clear-carry          ;  if IBF1 (PC1) = 0 (de-active) will jump to clear carry
    MOV DX, 0713h
    IN  AL , DX        ; read the pressed key from Pb
    STC                ;  set carry flag
   JMP  exit
clear-carry :    CLC  ;   clear carry flag
      exit      :
………………………..
```